

Tarea: 20 Ejercicios sobre Listas

1. Suma de elementos

Crea una función `suma_lista(lista)` que reciba una lista de números y retorne la suma de sus elementos.

Ejemplo:

`suma_lista([4, 1, 7, 3]) → 15`

2. Número mayor y menor

Crea una función `mayor_menor(lista)` que reciba una lista de números y retorne una tupla con el número mayor y el menor. No pueden usar `min` y `max`

Ejemplo:

`mayor_menor([10, 5, 22, 8]) → (22, 5)`

3. Contar elementos pares

Crea una función `contar_pares(lista)` que cuente cuántos números pares hay en una lista y retorne esa cantidad.

Ejemplo:

`contar_pares([3, 6, 2, 7, 10]) → 3`

4. Invertir lista

Crea una función `invertir_lista(lista)` que retorne una nueva lista con los elementos en orden inverso (sin usar `[::-1]` ni `reverse()`).

Ejemplo:

`invertir_lista([1, 2, 3, 4]) → [4, 3, 2, 1]`

5. Promedio de una lista

Crea una función `promedio(lista)` que retorne el promedio de los elementos de la lista como número decimal.

Ejemplo:

`promedio([4, 8, 6, 2]) → 5.0`

6. Buscar un elemento

Crea una función `buscar_elemento(lista, valor)` que retorne una lista con las posiciones donde aparece el valor en la lista. Si no se encuentra, retorna una lista vacía.

Ejemplo:

`buscar_elemento([5, 3, 7, 3, 9], 3) → [1, 3]`

7. Multiplicación de elementos

Crea una función `producto_lista(lista)` que retorne el resultado de multiplicar todos los elementos de la lista.

Ejemplo:

`producto_lista([2, 4, 3]) → 24`

8. Eliminar duplicados

Crea una función `eliminar_duplicados(lista)` que retorne una nueva lista sin duplicados, manteniendo el orden de aparición.

Ejemplo:

`eliminar_duplicados([1, 2, 2, 3, 1, 4]) → [1, 2, 3, 4]`

9. Filtrar elementos mayores a un valor

Crea una función `filtrar_mayores(lista, limite)` que retorne una nueva lista con los elementos mayores al valor dado.

Ejemplo:

`filtrar_mayores([2, 10, 5, 8], 6) → [10, 8]`

10. Concatenar listas elemento por elemento

Crea una función `concatenar_listas(lista1, lista2)` que retorne una nueva lista donde cada elemento sea la suma (numérica) o concatenación (si son cadenas) de los elementos de ambas listas.

Ejemplos:

`concatenar_listas([1, 2], [3, 4]) → [4, 6]`

`concatenar_listas(["a", "b"], ["x", "y"]) → ["ax", "by"]`

11. Contar cuántas veces aparece un valor

Crea una función `contar_valor(lista, valor)` que reciba una lista y un valor, y retorne cuántas veces aparece ese valor en la lista.

Ejemplo:

`contar_valor([1, 2, 2, 3, 2], 2) → 3`

12. Sumar solo los valores impares

Crea una función `suma_impares(lista)` que retorne la suma de todos los números impares en la lista.

Ejemplo:

`suma_impares([1, 4, 7, 10]) → 8`

13. Crear una lista de cuadrados

Crea una función `cuadrados(lista)` que retorne una nueva lista con los cuadrados de cada número de la lista original.

Ejemplo:

`cuadrados([1, 2, 3]) → [1, 4, 9]`

14. Eliminar elementos menores a un valor

Crea una función `eliminar_menores(lista, limite)` que retorne una nueva lista sin los elementos menores al límite dado.

Ejemplo:

`eliminar_menores([2, 10, 5, 1], 5) → [10, 5]`

15. Intercalar dos listas

Crea una función `intercalar_listas(lista1, lista2)` que reciba dos listas del mismo tamaño y retorne una nueva lista con los elementos intercalados.

Ejemplo:

`intercalar_listas([1, 3, 5], [2, 4, 6]) → [1, 2, 3, 4, 5, 6]`

16. Contar cuántos elementos son mayores al promedio

Crea una función `mayores_que_promedio(lista)` que retorne cuántos elementos son mayores al promedio de la lista.

Ejemplo:

`mayores_que_promedio([4, 6, 8, 2]) → 2`

17. Verificar si una lista está ordenada

Crea una función `esta_ordenada(lista)` que retorne `True` si los elementos están en orden ascendente, o `False` si no lo están.

Ejemplo:

`esta_ordenada([1, 2, 3, 4]) → True`

`esta_ordenada([1, 3, 2]) → False`

18. Obtener los valores únicos

Crea una función `valores_unicos(lista)` que retorne una lista con los elementos que aparecen solo una vez.

Ejemplo:

`valores_unicos([1, 2, 2, 3, 4, 4]) → [1, 3]`

19. Obtener la lista sin ceros

Crea una función `eliminar_ceros(lista)` que retorne una nueva lista sin los elementos que sean igual a 0.

Ejemplo:

`eliminar_ceros([0, 5, 0, 3]) → [5, 3]`

20. Repetir cada elemento n veces

Crea una función `repetir_elementos(lista, n)` que retorne una nueva lista donde cada elemento se repita `n` veces consecutivas.

Ejemplo:

`repetir_elementos([1, 2], 3) → [1, 1, 1, 2, 2, 2]`